



Open-Source Software Practice

Lab 05. Node and JavaScript

TA: Jiwon Choi (최지원, jasonchoi3@g.skku.edu)

Interactive Data Computing Lab (IDCLab)

College of Computing and Informatics,

Sungkyunkwan University

Goal



- Install Node.js
- Learn JavaScript Basics

Install Node.js



- Install with Package (.exe, .pkg, deb ...)
 - <https://nodejs.org/ko/download/>
- Install with Package Manager
 - Homebrew (Mac)
 - `brew install node`
 - apt (Ubuntu, Debian)
 - `curl -fsSL https://deb.nodesource.com/setup_lts.x`
 - `sudo -E bash - sudo apt-get install -y nodejs`

Installation Check



```
(base) x jasonchoi3 ~ node -v  
v18.0.0  
(base) jasonchoi3 ~ npm -v  
8.12.1  
(base) jasonchoi3 ~ |
```

Execute JavaScript Commands & Files with Node.js



- `ossp.js`

```
console.log("Hello World!")
```

- (test) `jasonchoi3` `~` `node` `ossp.js`
Hello World!

- (test) `jasonchoi3` `~` `node`
Welcome to Node.js v18.0.0.
Type ".help" for more information.
> `console.log("Hello World!")`
Hello World!
undefined
> █

Variables



- Declare variables with `let`, `const`

```
let year = 1398;  
let name = "SKKU";
```

```
const department = "Computer Science and Engineering";  
const ids = [20220001, 20220002, 20220003, 20220004, 20220005];
```

- `let` vs `const` ?
- `var` ?

Primitive Types



- Number, String, Boolean, undefined, null, and more..
- **Primitive types are immutable!**
 - Immutable: cannot be altered === cannot change

```
let string = "Sungkyunkwan University"  
string[0] = "B" // Error
```

Operators



- Arithmetic: +, -, *, /, %, ++, --
- Assignment: =, +=, -=, *=, /=, %=
- Comparison: ===, !==, >, <, >=, <=, ?
- Logical: &&, ||, !
- Bitwise: &, |, ~, ^, <<, >>
- Type: typeof
- Almost Same With C

Type Conversion



- How to Convert Type Explicitly?

```
typeof "123" // "string"  
typeof Number("123") // "number"
```

```
typeof 123 // "number"  
typeof String(123) // "string"
```

```
typeof (123).toString() // "string"
```

- Apply operators on variables of the same type

if statement



- if statement of C and JavaScript are same!

```
const a = 10;
const b = 20;

if (a > b){
    console.log("a is greater than b");
}
else if (a < b){
    console.log("a is less than b");
}
else{
    console.log("a is equal to b");
}
```

```
const university = "Sungkyunkwan University";

if (university === "Sungkyunkwan University")
{
    console.log("Welcome to SKKU");
}
else {
    console.log("You are not from SKKU");
}
```

for statement



- JavaScript has three for statement!
 - for: Default for statement (C-like)
 - for..in: Iterates with indices (or key of object)
 - for..of: Iterates with elements

```
const arr = ["Open", "Source", "Software"]
```

```
for (let i=0; i<3; i++) console.log(i, arr[i]) // Not int i=0;
```

```
for (let idx in arr) console.log(idx) // 1 2 3
```

```
for (let val of arr) console.log(val) // Open Source Software
```

while statement



- while statement of C and JavaScript are same!
- do...while also same.

```
let i = 0;
while (i < 3) {
    console.log(i);
    i++;
}
```

```
let i = 0;
do {
    console.log(i);
    i++;
} while (i < 3);
```

Array Data Structure



- Declare Array

```
const a = ["Open", "Source", "Software", 1398, ["Linux", "Windows", "MacOS"]];
```

- Array Length

```
a.length; // 5
```

- Typeof Array

```
typeof a // object  
Array.isArray(a) // true
```

Object Data Structure



```
const IDCLab = {  
  director: {  
    name: "Jaemin Jo"  
  },  
  students: [  
    { name: "John", id: 111 },  
    { name: "Zoey", id: 112 },  
    { name: "Chen", id: 113, graduated: true },  
  ]  
}
```

```
console.log(IDCLab.director)  
console.log(IDCLab.director.name)
```

```
console.log(IDCLab.students)  
console.log(IDCLab.students[0].name)
```

Object Data Structure



- Using Complex Data Structures

```
for (const student of IDCLab.students) {  
    if (student.graduated) console.log(student.name + " graduated")  
    else console.log(student.name + " is studying")  
}
```

- Everything in JavaScript Except Primitive Type is Object!

```
typeof [1,2,3] // "object"  
typeof {a:1, b:2} // "object"  
typeof function(){} // "function"...?
```

Object Quiz



- Object with const?

```
const IDCLab = {  
  director : "Jaemin Jo"  
}
```

```
IDCLab.director = "Jiwon Choi" // ?!?!?!
```

- Key with Phrase?

```
const IDCLab = {  
  director name : "Jaemin Jo", // ???  
}
```


Function



- Ways to Declare Functions

```
function sum(a, b) {  
    return a + b;  
}
```

```
const sum = function (a, b) { return a, b };
```

```
const sum = (a, b) => { return a + b };
```

```
const sum = (a, b) => a + b;
```

Array Methods



- Destructive Methods VS Non-Destructive Methods

```
const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
console.log(arr.push(11));  
console.log(arr.pop());  
console.log(arr.includes(2));  
console.log(arr.slice(2, 5));  
console.log(arr.splice(2, 5));  
console.log(arr.concat([11, 12, 13]));
```

```
console.log(arr.join(' '));  
console.log(arr.reverse());  
console.log(arr.shift());  
console.log(arr.unshift(1));  
console.log(arr.sort());  
console.log(arr.toString());
```

Array Methods



- Collective Operation Methods

```
const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
console.log(arr.map((item) => item * 2));  
console.log(arr.filter((item) => item % 2 === 0));  
console.log(arr.forEach((item) => {console.log(item * 2)}));  
console.log(arr.every((item) => item > 0));  
console.log(arr.some((item) => item > 10));
```

Array Methods



- Method Chaining

```
const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
arr.map(i => i * 2)  
    .filter(i => i % 3 === 0)  
    .forEach(i => console.log(i * i))
```

Summary Quiz



Write function that multiplies the number property.
(* Hint: Use for..in loop or Object.entries())

```
let menu = {  
  width: 200,  
  height: 300,  
  title: "My menu"  
};
```

→ multiplyNumeric(menu, 3); →

```
menu = {  
  width: 600,  
  height: 900,  
  title: "My menu"  
};
```

* multiplyNumeric returns nothing. Just modify menu object.

Homework Announcement



- JavaScript Practice (~10/21)
 - <https://www.codecademy.com/learn/introduction-to-javascript>
 - Details are on the i-Campus

Team Project Announcement



- Two or Three students are assigned to each team.
- Share your contacts (e.g., Phone number, Kakao talk ID) for your team project.
- You can check your teammates and send messages to share your contacts with teammates using i-Campus.